



PPG/PES

## RELATÓRIO FINAL

1. ACADÊMICO(S) PARTICIPANTE(S):

Antonio Carlos Contieri Júnior

2. ORIENTADOR: Prof. Msc. Thelma Elita Colanzi Lopes

3. DEPARTAMENTO: DIN

4. TÍTULO DO PROJETO:

Implementação de variabilidade da Linha de Produto de Software LPS-BET usando orientação a aspectos

5. INÍCIO: 01/05/2009

6. TÉRMINO: 12/02/2010

7. AVALIAÇÃO DO ORIENTADOR SOBRE O PIC.

O PIC propicia para qualquer acadêmico o contato com as atividades de pesquisa, que levam ao aprofundamento de estudos numa determinada área de pesquisa, ao amadurecimento profissional, à ampliação do raciocínio analítico, e a uma melhor preparação para a realização do Trabalho de Conclusão de Curso (TCC). Neste sentido, penso que este tipo de programa só tem a acrescentar à formação acadêmica dos universitários.

8. AVALIAÇÃO DO ORIENTADOR SOBRE O DESEMPENHO DO(S) ACADÊMICO(S)-PARTICIPANTE(S) NO PIC.

O acadêmico se dedicou na realização do projeto de pesquisa e mostrou uma evolução significativa durante o período. A dedicação do acadêmico foi excelente, inclusive os objetivos do projeto foram alcançados antes do tempo previsto. Portanto, considero muito bom o desempenho do acadêmico no desenvolvimento deste projeto.

9. AVALIAÇÃO DO(S) ACADÊMICO(S) SOBRE O PIC.

O projeto foi de extrema importância para o desenvolvimento do acadêmico envolvido, devido à experimentação, análise e publicação dos resultados envolvidos, proporcionando uma nova experiência e, além disso, trazendo uma contribuição para a comunidade acadêmica no que diz respeito a estudos envolvendo os temas principais deste trabalho: Linhas de Produto de Software e Programação Orientada a Aspectos.

<p><b>10. ESCANEAR E ANEXAR COMPROVANTE DE APRESENTAÇÃO DOS RESULTADOS DA PESQUISA EM EVENTOS CIENTÍFICOS. Se houve premiação, informar o evento e classificação e anexar comprovante.</b></p>		
<p><b>11. ESCANEAR E ANEXAR, SE HOUVER, COMPROVANTE DE PUBLICAÇÃO, COM A PARTICIPAÇÃO DO(S) ACADÊMICO(S), EM PERIÓDICOS INDEXADOS E/OU COM CORPO EDITORIAL.</b></p>		
 <p><b>UEM</b></p>		<p><b>PPG/PES</b></p>
<p><b>12. NORMAS PARA ELABORAÇÃO DO RELATÓRIO</b></p> <p>a) Capa  b) Folha de rosto  c) Resumo do trabalho  d) Sumário  e) Texto completo do trabalho  e.1) Introdução  e.2) Objetivos  e.3) Desenvolvimento (Materiais e Métodos)  e.4) Resultados e Discussão  e.5) Conclusões  f) Bibliografia (de acordo com as normas da ABNT)  g) Anexos (referentes à pesquisa): comprovantes de apresentação dos resultados da pesquisa em eventos científicos, e/ou comprovante de publicação, com a participação do(s) acadêmico(s) em periódicos indexados e/ou com corpo editorial (conforme o regulamento do Programa PIC)</p>		
<p><b>13. OUTRAS INSTRUÇÕES</b></p> <ul style="list-style-type: none"> <li>• O resumo, não excedendo 150 palavras, deverá conter informações sobre o objetivo da pesquisa, os materiais experimentais, os métodos empregados, resultados e conclusões. Até seis palavras-chave deverão ser acrescentadas no final do resumo;</li> <li>• O relatório não deverá exceder a 50 páginas digitadas, incluindo figuras, tabelas e referências bibliográficas;</li> <li>• Deverão ser escritos em espaço 1,5 e ter suas páginas numeradas;</li> <li>• Deverá ser utilizada, no corpo do texto, letra tipo Times New Roman, tamanho 12;</li> <li>• Os parágrafos deverão ser recuados em 1,25cm na primeira linha e o alinhamento deverá ser justificado;</li> </ul> <p>As tabelas, figuras, gráficos etc. deverão estar inseridas nos seus respectivos locais dentro do texto.</p>		

UNIVERSIDADE ESTADUAL DE MARINGÁ  
PROGRAMA DE INICIAÇÃO CIENTÍFICA – PIC  
DEPARTAMENTO DE INFORMÁTICA  
ORIENTADORA: Prof<sup>a</sup>. Msc. Thelma Elita Colanzi Lopes  
ACADÊMICO: Antonio Carlos Contieri Júnior

**IMPLEMENTAÇÃO DE VARIABILIDADE DA LINHA DE  
PRODUTO DE SOFTWARE LPS-BET USANDO  
ORIENTAÇÃO A ASPECTOS**

Maringá, 11 de Fevereiro de 2010.

UNIVERSIDADE ESTADUAL DE MARINGÁ  
PROGRAMA DE INICIAÇÃO CIENTÍFICA – PIC  
DEPARTAMENTO DE INFORMÁTICA  
ORIENTADORA: Prof<sup>a</sup>. Msc. Thelma Elita Colanzi Lopes  
ACADÊMICO: Antonio Carlos Contieri Júnior

## **IMPLEMENTAÇÃO DE VARIABILIDADE DA LINHA DE PRODUTO DE SOFTWARE LPS-BET USANDO ORIENTAÇÃO A ASPECTOS**

Relatório contendo os resultados finais do projeto de  
iniciação científica vinculado ao Programa PIC-UEM.

Maringá, 11 de Fevereiro de 2010.

## RESUMO

O desenvolvimento de Linhas de Produtos de Software (LPS) é uma das técnicas mais utilizadas atualmente para viabilizar o reuso de software. Neste sentido, vários trabalhos vêm sendo realizados com o intuito de aprimorar essa atividade. Algumas pesquisas têm mostrado que a implementação de variabilidades de LPS usando aspectos pode ser benéfica em várias situações. Sendo assim, este projeto teve por objetivo implementar uma variabilidade de uma LPS para gestão de bilhetes eletrônicos de transporte usando orientação a aspectos, avaliando as vantagens e desvantagens desta implementação para este caso específico. Para isso foi realizado um estudo bibliográfico sobre os assuntos envolvidos e, na fase de implementação, foram utilizados vários recursos tecnológicos como Java, AspectJ e Hibernate. Este trabalho também apresenta um conjunto de métricas utilizadas na avaliação desta implementação, como forma de embasar os resultados obtidos. Esta pesquisa concluiu que, para arquiteturas baseadas em componentes, a implementação por meio de aspectos não favoreceu a evolução da LPS.

Palavras-chave: Linha de Produto de Software, Programação Orientada a Aspectos, Desenvolvimento baseado em Componentes.

## SUMÁRIO

1 – INTRODUÇÃO .....	7
2 – OBJETIVOS.....	8
3 – DESENVOLVIMENTO (MATERIAIS E MÉTODOS) .....	8
4 – RESULTADOS E DISCUSSÃO .....	10
5 - CONCLUSÃO .....	14
BIBLIOGRAFIA .....	15

## 1 – INTRODUÇÃO

As empresas de desenvolvimento de software buscam constantemente por alternativas de trabalho que permitam aumentar o reuso de artefatos de software e reduzir o time-to-market. Isto gera a necessidade de técnicas que permitam realizar o reuso de uma maneira sistemática e previsível. O reuso sistemático de software demanda uma mudança da abordagem de construção de sistemas únicos para o desenvolvimento de famílias de sistemas. Uma das abordagens para reuso sistemático é linha de produto de software (LPS).

A abordagem de LPS visa à geração de produtos específicos para um determinado domínio baseado no reaproveitamento dos ativos centrais (*core assets*), que tem um conjunto de características comuns, bem como um conjunto de características variáveis, o que permite adiar algumas decisões de projeto [van der Linden, Schmid, Rommes, 2007]. Os produtos específicos são originados da composição e da configuração desses ativos. Em outras palavras, pode-se dizer que entre os produtos de uma LPS há um conjunto comum de requisitos e há também variabilidades significativas entre outros requisitos. Por exemplo, uma família de sistemas para gestão de bilhetes eletrônicos de transporte possui um núcleo de funcionalidades comum tal como o uso de um cartão plástico para pagamento de corridas. Ela possui ainda características variáveis como o tipo de integração entre corridas que funciona de forma diferente entre os diversos produtos que podem ser gerados para esta LPS.

Essas características variáveis entre os produtos de uma LPS são denominadas de variabilidades. Elas são descritas em termos de pontos de variação e variantes. Um ponto de variação é um local específico no artefato da LPS ao qual uma decisão de projeto está associada. Cada ponto de variação, por sua vez, está associado a um conjunto de variantes que correspondem às alternativas de projeto para uma variabilidade.

Clements *et al.* (2006) afirmam que a organização de LPS engloba três atividades principais: desenvolvimento do núcleo de artefatos, desenvolvimento do produto e gerenciamento da LPS. A atividade de desenvolvimento do núcleo de artefatos pode ser chamada de engenharia de domínio, assim como a atividade de desenvolvimento do produto pode ser chamada de engenharia de aplicação. O núcleo de artefatos (do inglês *core assets*) é constituído dos recursos comuns a toda a LPS e pode incluir planos, requisitos, projetos, documentação, testes e código.

O desenvolvimento de LPS pode usar várias técnicas que facilitam a reutilização, como *frameworks*, componentes, padrões de projeto e linguagens orientadas a aspectos. E vários trabalhos enfatizam alternativas distintas para a implementação de variabilidades no contexto de LPS. Nesse contexto, Donegan e Masiero (2007) discutem questões sobre projeto de famílias de produtos. Para

ilustrar essas questões foi desenvolvida a LPS-BET, uma linha de produtos de software para gestão de bilhetes eletrônicos de transporte. Foi proposta uma adaptação do processo ESPLEP (Gomaa, 2004) e em cada um dos quatro incrementos foram construídos, respectivamente, o núcleo da LPS e cada uma das três aplicações-referência da LPS (os produtos para as cidades de Fortaleza/CE, Campo Grande/MS e São Carlos/SP). A arquitetura da LPS-BET é baseada em componentes do tipo caixa-preta. No entanto, Donegan (2008) discute a possibilidade de uso de Programação Orientada a Aspectos (POA) para a representação de requisitos não-funcionais e de variabilidades dessa LPS.

Sucintamente, aspectos são módulos que permitem separar e encapsular interesses transversais eliminando o espalhamento e o entrelaçamento de código. O aspecto adiciona uma funcionalidade ao código-base interceptando o fluxo de execução. No código-base não é preciso haver menção explícita ao aspecto. Portanto, ao utilizar POA, os sistemas ficam mais legíveis, fáceis de entender, implementar, integrar, reusar, personalizar, evoluir e manter (Kiczales, 1997).

## **2 – OBJETIVOS**

Considerando os estudos sobre utilização de aspectos para implementação de variabilidades de LPS, os objetivos deste projeto consistem: (i) da implementação de uma variabilidade da LPS-BET usando aspectos; e, (ii) da comparação dos resultados obtidos da implementação da variabilidade utilizando POA com a implementação anterior da LPS-BET.

## **3 – DESENVOLVIMENTO (MATERIAIS E MÉTODOS)**

Com a conclusão de uma revisão sistemática sobre trabalhos nos quais se têm utilizado aspectos para a representação de variabilidades de LPS foram selecionados trabalhos científicos que serviram de fonte para o estudo bibliográfico que foi realizado no presente projeto.

Na seqüência, estudou-se a especificação e a implementação da versão existente da LPS-BET e as tecnologias utilizadas na implementação dessa LPS. Esta versão não utiliza POA para a implementação de variabilidades e é baseada em componentes do tipo caixa-preta, aqueles cujo acesso se dá somente pela sua interface. Posteriormente, foi escolhida a variabilidade de Pagamento de Cartão da referida LPS que foi, então, desenvolvida utilizando POA. Essa implementação com aspectos foi feita considerando as recomendações apresentadas no trabalho de Figueiredo *et al.* (2008). Neste trabalho, foram realizados diversos experimentos em duas LPS. Para cada uma delas variabilidades foram implementadas com e sem o uso de aspectos. Em seus experimentos, foram estabelecidas uma série de métricas a fim de observar vantagens e desvantagens de se utilizar POA



nesse contexto. Ao final das atividades, eles obtiveram evidências de que, em alguns casos, os aspectos de fato favorecem a evolução da LPS.

O desenvolvimento do aspecto para a referida variabilidade foi realizada seguindo a metodologia utilizada por Donegan (2008). Para que a decisão de projeto adotada para a criação do aspecto possa ser compreendida, é preciso entender a nomenclatura adotada por esta autora para a definição dos componentes da LPS-BET. Os componentes com terminação *Mgr* são considerados de negócio (Chessman e Daniels, 2001 apud Donegan, 2008, p. 61), e aqueles com terminação *Ctrl*, de controle (Gomma, 2004 apud Donegan, 2008, p. 61). Sobre a relação entre aspectos e componentes Donegan (2008) salienta que:

Os componentes de negócio são extraídos do refinamento do modelo conceitual da mesma forma como são extraídos no projeto sem o uso de aspectos. Os aspectos substituem componentes controladores e de sistema. Eles entrecortam componentes básicos, adicionam lógicas de sistema ou de controle e obtêm informações necessárias de componentes de negócio. Dessa forma, os aspectos atuam como uma espécie de código de ligação (*glue code*) entre o núcleo e a variabilidades da LPS. (Donegan, 2008; p. 98)

Além disso, ela fornece as seguintes diretrizes para a utilização de aspectos na arquitetura da LPS-BET:

Para soluções de projeto baseadas em componentes em que é necessário substituir um componente A' do núcleo por outro B' com a variabilidade a ser implementada, a solução usando aspectos requer que o aspecto entrecorte as operações das interfaces do componente A', não permitindo que as operações do componente A' sejam executadas. Pode ser feita uma solução em que o aspecto chama as operações do componente B' (componente de negócio) ou implementa as operações que estariam no componente B' (componente de controle ou de sistema). O aspecto executaria de modo a não permitir que a aplicação-referência utilize as operações do componente do núcleo (componente A'). Mesmo que o componente não seja utilizado, o componente se encontra conectado aos outros componentes e a arquitetura da aplicação-referência não corresponde à arquitetura de fato, podendo dificultar manutenções posteriores na aplicação. Nesse ponto a solução usando componentes possui vantagem sobre a solução usando aspectos, pois a arquitetura dos componentes corresponde realmente à forma como a aplicação funciona. (Donegan, 2008, p. 99)

Desta maneira, o desenvolvimento do aspecto foi realizado à luz destas diretrizes, conforme descrição na próxima seção deste relatório. Para viabilizar esta atividade foram utilizados recursos tecnológicos como o sistema operacional Windows XP e algumas ferramentas de programação e modelagem. Para a criação dos diagramas utilizou-se Jude/Community (Jude, 2009). A codificação do projeto seguiu o modelo de Donegan (2008), utilizando a IDE Eclipse (Eclipse IDE, 2009), linguagem Java e AspectJ (AspectJ, 2009), banco de dados PostgreSQL (PostgreSQL, 2009), Apache Velocity (Velocity, 2009) para geração da interface de usuário, Subversion (Subversion, 2009) para controle de versões, Maven 2 (Maven 2, 2009) para construção do projeto e controle das bibliotecas utilizadas e alguns *frameworks* de apoio, como Hibernate (Hibernate, 2009) para

persistência e Spring (Spring, 2009) para diversas funções, como a configuração dos componentes e aspectos, controle de transações, entre outras.

O desenvolvimento da LPS-BET é constituído basicamente de componentes comuns a vários produtos gerados para um domínio de transporte público, que compõem o núcleo da aplicação, e de componentes específicos para cada variabilidade. O *framework* Spring utilizado fornece meios de identificar quais componentes são utilizados e como eles se ligam uns aos outros, por meio de interfaces (Donegan, 2008 – p. 63).

O *framework* Hibernate foi utilizado na camada de persistência da LPS-BET, a fim de mapear os objetos para um banco de dados relacional. Cada uma das classes que identifica um objeto que deve ser persistido possui um arquivo de configuração que representa seu mapeamento no banco de dados. Este arquivo contém informações dos campos da classe e suas relações com outros objetos. A manipulação das informações persistidas pelo sistema é feita basicamente por classes com terminação DAO. Estas classes possuem referência a um objeto denominado **HibernateDaoSupport**, que faz parte do Spring. Por meio dessa integração entre Spring e Hibernate é possível tratar facilmente as operações da camada de persistência, uma vez que este objeto provê acesso a várias funcionalidades, como a sessão atual da aplicação, e métodos para criar, alterar e excluir os objetos mapeados.

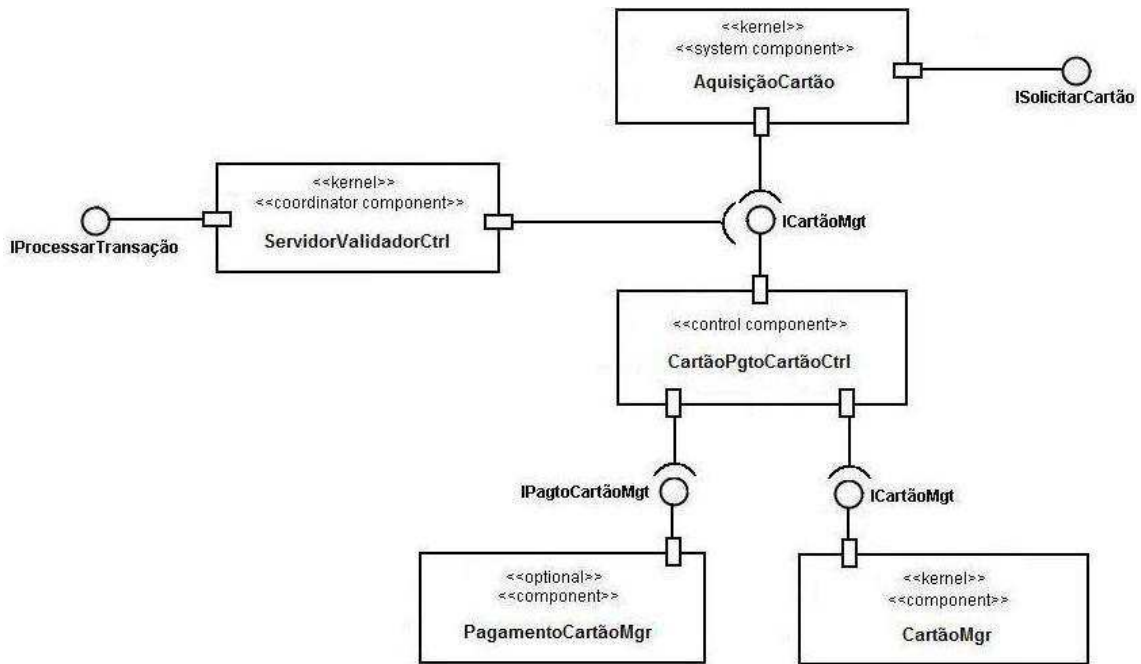
A utilização da IDE Eclipse facilita o desenvolvimento em diversas etapas como a criação de classes e aspectos, edição de arquivos do Velocity, gerenciamento de banco de dados, controle de versões, controle de dependências do projeto, entre outros. Algumas dessas funcionalidades não estão presentes inicialmente no Eclipse, é preciso instalar e configurar alguns *plugins* para aumentar a integração das ferramentas.

Para a utilização do repositório e configuração de cada tecnologia supracitada foram seguidos os tutoriais disponíveis em Donegan (2008) e Meilsmith (2008).

#### **4 – RESULTADOS E DISCUSSÃO**

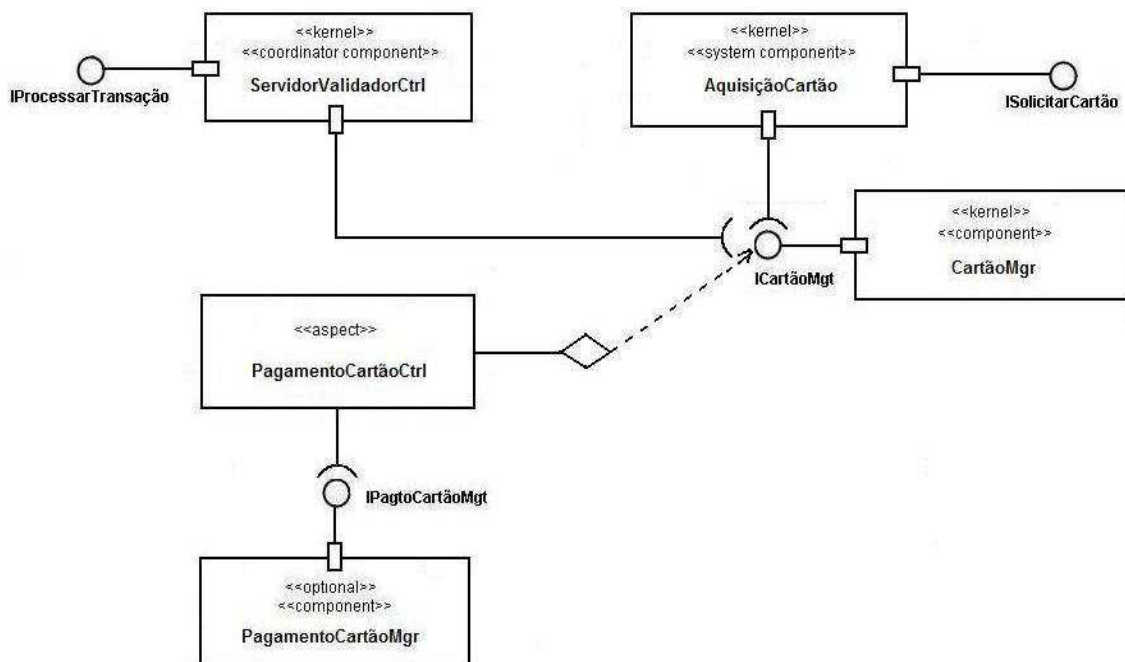
A LPS-BET é uma LPS para gestão de bilhetes eletrônicos de transporte que possui como uma de suas características variáveis o uso de um cartão eletrônico para pagamento de corridas, o que constitui a variabilidade Pagamento de Cartão. A solução usando componentes caixa-preta feita por Donegan (2008) foi a de projetar inicialmente o componente `CartãoMgr` de tal modo que ele pudesse ser reusado por todas as aplicações. Essa solução é mostrada na Figura 1. O componente `CartãoPgtoCartãoCtrl` fornece a interface `ICartaoMgt` para o componente `ServidorValidadorCtrl` e para o componente `AquisiçãoCartão`. O componente

CartãoPgtoCartãoCtrl requer a interface IPagtoCartãoMgt e ICartãoMgt para realizar o pagamento de cartões. Ele é usado apenas quando existir o Pagamento de Cartão, o que somente ocorre na aplicação-referência de Fortaleza.



**Figura 1: Solução utilizada para integrar a característica Pagamento de Cartão na arquitetura da LPS-BET (Donegan, 2008).**

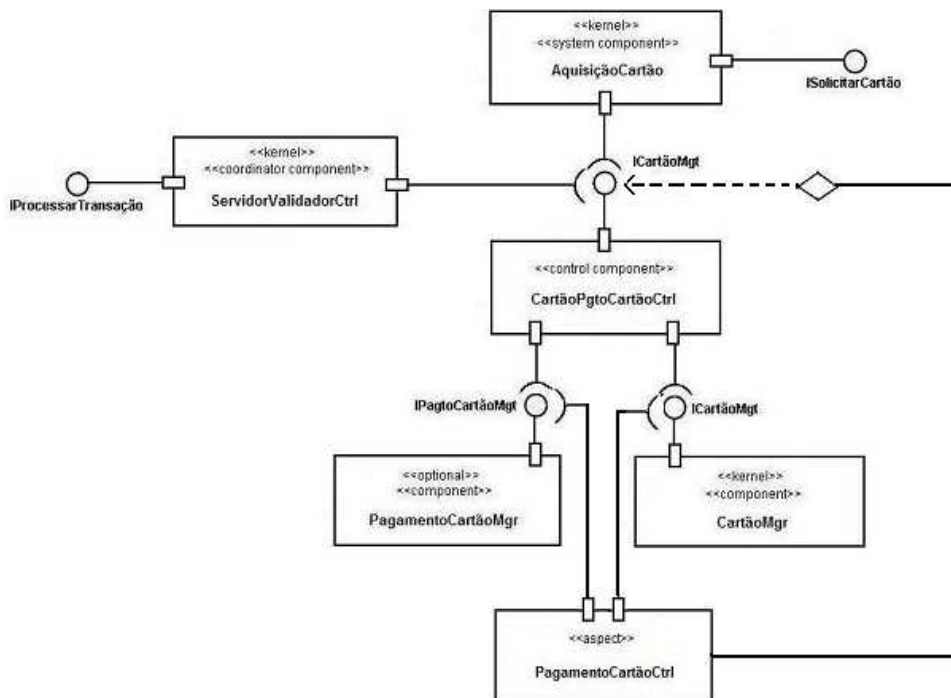
Para a utilização de aspectos na implementação da mesma característica, foi seguido o projeto proposto por Meilsmith (2008), e apresentado na Figura 2.



**Figura 2: Uma solução usando aspectos para adicionar a característica Pagamento de Cartão na arquitetura (Meilsmith, 2008).**

A principal diferença entre os dois projetos é que os componentes básicos ServidorValidadorCtrl e AquisicaoCartao não precisariam requerer a interface ICartaoMgt do componente CartãoPgtoCartãoCtrl. Portanto, esse componente de controle seria excluído, pois o aspecto PagamentoCartãoCtrl implementaria todas as operações que estão no componente CartãoPgtoCartãoCtrl. O aspecto PagamentoCartãoCtrl entrecortaria todas as operações ICartaoMgt e requereria as operações da interface IPagtoCartãoMgt do componente de negócio PagamentoCartãoMgr. Os outros componentes não deveriam chamar CartãoPgtoCartãoCtrl e sim CartaoMgr.

Contudo, foram encontradas algumas limitações em relação ao projeto proposto na Figura 2. Na prática, o aspecto requer uma interface do tipo ICartaoMgt, além do IPagtoCartaoMgt, pois, para substituir o componente de controle CartãoPgtoCartãoCtrl e assumir o controle da operação, alguns métodos daquela interface devem ser chamados. Desta forma, o componente CartãoPgtoCartãoCtrl não seria excluído totalmente, porém o aspecto assumiria o controle cada vez que um método deste componente fosse chamado e executaria a operação contida neste aspecto. Pode-se dizer que o componente de controle passou a ser apenas figurativo neste caso. Assim, os componentes ServidorValidadorCtrl e AquisicaoCartao continuariam a requerer a interface ICartaoMgt do componente CartaoPgtoCartaoCtrl, e não do CartaoMgr como havia sido proposto. A Figura 3 demonstra o projeto final do aspecto.



**Figura 3: Solução utilizada para adicionar o aspecto PagamentoCartãoCtrl na arquitetura da LPS-BET.**

Para avaliar a implementação realizada e comparar a implementação usando POA com a implementação usando componentes, algumas métricas foram definidas. Como os aspectos utilizam os componentes de negócio já existentes, não foi notada nenhuma grande diferença nos resultados. O Quadro 1 apresenta as métricas obtidas do desenvolvimento da variabilidade Pagamento de Cartão.

**Quadro 1: Métricas obtidas da variabilidade Pagamento de Cartão**

	<b>Métrica</b>	<b>Valor</b>
<b>Componente</b>	Interfaces envolvidas com a variabilidade	2
	Componentes envolvidos com a variabilidade	5
	Aspectos envolvidos com a variabilidade	1
	Linhas de código do componente substituído	193
	Linhas de código dos métodos do componente substituído	132
<b>Aspecto</b>	Linhas de código do aspecto	259
	Adendos por aspecto	32
	Linhas de código dos adendos do aspecto	98
	Interfaces interceptadas por aspecto	1
	Operações da interface interceptadas	32
	Classes que referenciam o componente substituído por aspecto	4

Analisando as métricas apresentadas no quadro anterior, pode-se verificar que não existe uma diferença significativa nas implementações, pois o aspecto apenas utiliza o componente de controle em sua construção.

Apesar da diferença de linhas de código entre o componente substituído (`CartaoPgtoCartaoCtrl`) e o aspecto (`PagamentoCartaoCtrl`) ser substancial, esta diferença pode ser reduzida se os pontos de junção e adendos do aspecto forem formatados de forma semelhante à declaração de um método do componente, pois para um melhor entendimento de código, eles foram separados utilizando um número maior de linhas. O mesmo princípio pode ser usado para explicar a diferença de número de linhas dos métodos do componente e o número de linhas dos adendos do aspecto. Outro ponto que mostra que não houve diferenças significativas está relacionado ao número de operações da interface da variabilidade e o número de adendos do aspecto, já que ambos são iguais, devido ao fato do aspecto possuir um adendo para cada método da interface.

Além disso, como nenhuma alteração no funcionamento da variabilidade foi necessária, o aspecto serviu apenas como ligação entre as requisições aos métodos da interface da variabilidade e

sua implementação concreta. Isso faz com que ele assuma o controle das funções do componente de negócio, mas ainda o referencie por meio da chamada de seus métodos, não possibilitando a exclusão do mesmo.

Assim, os componentes de controle originais não são mais utilizados por outras partes da arquitetura, porém ainda permanecem conectados. Portanto, a utilização de aspectos não trouxe benefícios visíveis para a LPS-BET na substituição do componente pelo aspecto proposto. Donegan (2008, p. 99) havia mencionado que a solução usando componentes possui vantagem sobre a solução usando aspectos, devido à arquitetura baseada em componentes corresponder realmente à forma como a aplicação funciona. Assim, os fatos observados na realização deste projeto ratificam que não é vantajosa a substituição dos componentes de controle da arquitetura da LPS-BET por aspectos.

## **5 - CONCLUSÃO**

Este trabalho foi baseado na utilização de POA para a implementação de uma variabilidade de uma LPS já existente, a LPS-BET, que apresenta uma arquitetura baseada em componentes caixa-preta. Com o objetivo de analisar e realizar um estudo comparativo entre a utilização de aspectos e de componentes em um mesmo contexto, um aspecto referente à variabilidade de Pagamento de Cartão foi desenvolvido, substituindo assim o componente controlador e analisando os resultados.

A utilização de aspectos visa melhorar uma implementação fornecendo essencialmente duas características: modularização e reutilização. No presente trabalho, do ponto de vista técnico, o aspecto é facilmente mantido, porém, sua inclusão na LPS-BET altera a arquitetura dos componentes, que continuam interligados, mas não são de fato utilizados. Ainda, o aspecto mantém a modularidade do sistema, pois assume o papel realizado anteriormente pelos componentes.

De uma forma geral, o aspecto implementado como forma de substituir o componente controlador da variabilidade analisada da LPS-BET não proporcionou grandes impactos na melhoria da arquitetura. Além disso, o próprio projeto inicial do aspecto precisou de modificações para o seu funcionamento. Caso diversas variabilidades sejam substituídas por aspectos, o excesso de adaptações pode gerar problemas de manutenção, prejudicando tanto o funcionamento como o entendimento da aplicação como um todo.

Finalmente, conforme foi relatado, este trabalho concluiu que para um sistema projetado e inicialmente desenvolvido com base em componentes caixa-preta, a implementação de variabilidades por meio de aspectos pode não ser tão produtiva quanto o esperado. Os melhores

resultados só serão obtidos caso os aspectos não interfiram na estrutura gerada pelos componentes, ou seja, quando eles promovem uma extensão dos componentes, ao invés de substituí-los como um todo.

Dentre os trabalhos futuros está prevista a geração automática dos produtos da LPS-BET usando a nova versão da implementação da variabilidade desenvolvida neste trabalho usando POA, por meio do gerador de aplicação Captor (SHIMABUKURO, 2006), assim como Donegan (2008) fez em seu trabalho. Para isso é preciso criar uma nova configuração de entrada no Captor, que não estava no escopo deste trabalho.

Ainda, é possível realizar novos estudos de caso da LPS-BET para reforçar as evidências da utilização de aspectos. Outro trabalho possível é realizar uma evolução da LPS-BET, onde seriam implementadas novas variabilidades utilizando POA e componentes. Dessa forma, as variabilidades seriam as mesmas, possibilitando a comparação dos resultados alcançados entre as abordagens.

## **BIBLIOGRAFIA**

ASPECTJ. Disponível para acesso na URL: <http://www.eclipse.org/aspectj/>. Acesso em 18 de dezembro de 2009.

CLEMENTS, P. C., JONES, L. G., MCGREGOR, J. D., NORTHROP, L. M., Getting There From Here: A Roadmap for Software Product Line Adoption. **Communications of the ACM**. Vol. 49, Nº 12. December, 2006.

DONEGAN, Paula Marques; MASIERO, P. C. . Design Issues in a Component-based Software Product Line. In: SIMPÓSIO BRASILEIRO DE COMPONENTES, ARQUITETURAS E REUSO DE SOFTWARE, 2007, Campinas. **Anais....** Porto Alegre : Sociedade Brasileira de Computação, 2007. v. 1. p. 3-16.

DONEGAN, Paula Marques. **Geração de Famílias de Produtos de Software com Arquitetura Baseada em Componentes**. 2008. Dissertação (Mestrado em Ciência da Computação e Matemática Computacional). ICMC-USP/São Carlos. São Carlos, SP.

ECLIPSE. **The Eclipse Foundation**. Disponível em: <http://www.eclipse.org/>. Acessado em janeiro/2009.

ELER, M. M. **Um Método para o Desenvolvimento de Software baseado em Componentes e Aspectos**. Dissertação de Mestrado, ICMC, Universidade de São Paulo (USP), 2006.

FIGUEIREDO, E., CACHO, N., SANT'ANNA, C., MONTEIRO, M., KULESZA, U., GARCIA, A., SOARES, S., FERRARI, F., KHAN, S., FILHO, F., DANTAS, F. Evolving Software Product Lines with Aspects: An Empirical Study on Design Stability. In: 30th INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, **Proceedings....**, p. 261-270, Leipzig, Germany 2008.

GOMAA, H. **Designing Software Product Lines with UML**. Addison-Wesley: Boston, 2004.

HIBERNATE. **Hibernate - Relational Persistence for Java and .NET**. Disponível em: <http://www.hibernate.org/>. Acessado em janeiro/2009.

JUDE/COMMUNITY - Free UML Modeling Tool. Disponível para acesso na URL: <http://jude.change-vision.com/jude-web/index.html>. Acesso em 18 de dezembro de 2009.

KICZALES, G. et al. **Aspect-Oriented Programming**. European Conference on Object-Oriented Programming (ECOOP), LNCS (1241), Springer-Verlag, Finland., June 1997.

KICZALES, G.; HILSDALE, E.; HUGUNIN, J.; KERSTEN, M.; PALM, J.; GRISWOLD, W. An Overview of AspectJ. 15TH EUROPEAN CONFERENCE ON OBJECT-ORIENTED PROGRAMMING, **Proceedings...** p. 327–353, 2001.

MAVEN 2. **Apache Maven Project**. Disponível para acesso na URL: <http://maven.apache.org/>. Acesso em 01 de dezembro de 2009.

MEILSMITH, Guilherme Antoniali. **Uma investigação quanto ao uso de aspectos para implementação de variabilidades de Linha de Produto de Software**. 2008. Trabalho de Graduação (Bacharelado em Ciência da Computação). DIN-UEM. Maringá, PR.

SHIMABUKURO, E. K. J. **Um Gerador de Aplicações Configurável**. Dissertação de Mestrado, ICMC, Universidade de São Paulo (USP), 2006.

SPRING. **Spring Framework**. Disponível para acesso na URL: <http://www.springsource.org>, em 18/12/2009.

VELOCITY. **Apache Velocity Project**. Disponível na URL: <http://velocity.apache.org/>. Acesso em 19 de dezembro de 2009.

van der LINDEN, F. SCHMID, K. ROMMES, E. **Software Product Lines in Action – The Best Industrial Practice in Product Line Engineering**, Springer, 2007.



Anexo – Comprovante de publicação de resultados parciais do PIC.

**ESELAW'09**  
VI Experimental Software  
Engineering Latin  
American Workshop



**São Carlos  
Capital da Tecnologia**

## Certificado

Certificamos que  
**Antonio Carlos Contieri Júnior**

apresentou o artigo «Implementação de Variabilidades de Linha de Produto de Software utilizando Aspectos: Uma Revisão Sistemática» durante o VI Experimental Software Engineering Latin American Workshop realizado nos dias 11, 12 e 13 de novembro de 2009, no Departamento de Computação da Universidade Federal de São Carlos-SP.

São Carlos, 13 de novembro de 2009.

  
\_\_\_\_\_  
Prof.ª. Dra. Sandra Camargo Pinto Ferraz Fabbri  
Coordenadora Geral

  
\_\_\_\_\_  
Prof. Dr. Auri Marcelo Rizzo Vincenzi  
Coordenador do Comitê de Programa

---

